# 利用自适应假近邻方法优化卷积神经网络的图像目标分类算法 *

黄　珍 [1a]，李志浩 [2]，苑　毅 [1b]，阮文惠 [1a]

(1. 兰州文理学院 a. 数字媒体学院；b. 传媒工程学院，兰州 730000；2. 兰州大学 信息科学与工程学院，兰州 730000)

**摘　要：**为了能够在处理不同的数据类型或任务时得到良好的结果，设计了基于自适应假近邻方法的卷积神经网络(CNN)架构。将中心矩的思想应用在 CNN 的池化操作中，利用稀疏滤波算法实现训练过程的无监督化，并设置 CNN 算法的卷积掩模(卷积核)的大小和每层卷积单位(CNN 神经元)的数量；此外，该架构还利用自适应假近邻方法实现简化建模和预测等任务。实验结果证实，提出的改进 CNN 架构的复杂度较低，它可以更快地接受训练并且不易产生过度拟合。

**关键词：**卷积神经网络；假近邻；目标分类；矩池化；稀疏滤波

**中图分类号：**TP391.4　　doi: 10.3969/j.issn.1001-3695.2018.05.0339

## Image target classification algorithm based on adaptive artificial near-neighbor method for convolution nerve network optimization

Huang Zhen[1a], Li Zhihao[2], Yuan Yi[1b], Ruan Wenhui[1a]

(1. *a. School of Digital Media*, *b. School of Media Engineering*, *Lanzhou University of Arts & Sciences*, *Lanzhou* 73000, *China*; 2. *School of Information Science & Engineering*, *Lanzhou University*, *Lanzhou* 730000, *China*)

**Abstract:** In order to obtain good results when dealing with different data types or tasks, this paper designed a convolutional neural network (CNN) architecture based on adaptive false nearest neighbors method. It applied the center moment to CNN pooling operation and it used sparse filtering algorithm to realize the unsupervised training process, and it also set the size of the convolution mask (convolution kernel) of the CNN algorithm and the unit of convolution (CNN neural) . In addition, the architecture also used adaptive false nearest neighbors method to simplify tasks such as modeling and prediction. Experimental results confirm that the improved CNN architecture is less complex and it can be trained faster and it is less prone to overfitting.

**Key words:** convolutional neural network; false nearest neighbors; target classification; moment pooling; sparse filtering



## 0　引言

深度学习(deep learning，DL)[1]算法已成为分析海量数据的主要方法，它能够提取有意义的特征，以帮助理解和解决复杂的大数据处理问题。DL 算法在多个层次上运行，每个层次由一组涉及线性和非线性组件的回归模型组成，多种模型的组合使复杂功能的表示成为可能。大多数 DL 算法类似于人工神经网络，输入向量在包含操作单元的连续层中被处理以强调或抑制特征。目前，DL 算法主要应用在数据分类[2]、图像识别[3]、机器翻译[4]等方面。

卷积神经网络(convolutional neural network，CNN)[5]是一种经典的 DL 算法，它能够从音频和视频等时间相关的数据中提取出特定的特征。然而，CNN 的性能在很大程度上取决于它的体系结构，包括层数、每层单位和卷积掩模大小。而且，在处理不同的数据类型或任务时，适用于特定问题的体系结构可能并不通用。解决这个问题的常用替代方法是评估不同的 CNN 架构，以选择性能最好的架构。这样的过程显然具有许多缺点：a)对单个架构的训练和评估已经是计算密集型的，因此对许多场景的总体评估可能是不可行的；b)由经验定义的体系结构可能不适合所考虑的问题，因此可能无法获得理想的结果。

通常，用于避免评估多个 CNN 设置的策略需要考虑额外的训练阶段来调整与卷积单元相关的权重。这也增加了计算负

**收稿日期：**2018-05-07；**修回日期：**2018-07-04　　**基金项目：**甘肃省自然科学基金资助项目（17JR5RA007）；甘肃省教育厅资助项目（2016A-104）；兰州文理学院种子基金（自然）资助项目（17XJZZ005）；国家自然科学基金资助项目（61403329）

**作者简介：**黄珍（1980-），女，湖北黄冈人，副教授，硕士，主要研究方向为图像处理、模式识别（zhang838146chen032@163.com）；李志浩（1978-），男，山西大同人，讲师，博士研究生，主要研究方向为智能计算、生物信息学；苑毅（1979-），男，河北安平人，副教授，硕士，主要研究方向为电子技术、智能信息系统；阮文惠（1964-），女，吉林吉林人，教授，本科，主要研究方向为计算机应用、图形图像.

担，并且需要数千/数百万个例子才能产生合理的结果[6]。另一种策略是使用 CNN 集合，它通常允许利用最大数量的迭代次数，但需要花费更多的体系结构来获得相关结果。

本文提出了一种新的 CNN 体系结构。它将中心矩的思想应用在 CNN 的池化操作中，利用稀疏滤波算法实现训练过程的无监督化。受假近邻(False Nearest Neighbors，FNN)[7]方法的启发，本文的方法分析了每个 CNN 层的卷积操作产生的输入和输出图像，以估计卷积掩模的适当尺寸和每层卷积单元的数量，并利用自适应假近邻方法实现简化建模和预测等任务。

## 1　相关研究

CNN 算法被广泛应用在分类问题中。文献[8]使用 Maxout 模型和退出策略来改进 CNN 特征提取。Maxout 是一个激活函数，它将卷积运算 $x$ 的结果乘以一个对角权重矩阵 $W$，并根据表达式 $z = x^T W + b$ 将其与一个偏差向量 $b$ 相加。$z$ 中的最大值是输出，该模型的错误率为 0.45%。文献[9]中提出了具有两级特征的多尺度 CNN(MS-CNN)分类器。文献[10]研究了稀疏网络(SNoW)表示的图像边缘并识别来自哥伦比亚大学图像库(COIL-100)的物体。SNoW 中的每个单元都代表一个负责将对象分类到特定类的子网络，因此SNoW 包含与标签一样多的单元。文献[11]中提出了一个新的框架，它能够减少在训练期间花费的时间，并使用反卷积网络来评估训练行为。采用相关性分析输入图像与重建图像之间的相似度，作为适应度函数。最后，应用 Nelder-Mead 方法来优化适应度函数并找到最佳结构。该框架为参数提供了一个范围，并为分析的数据集提供了足够的准确度。文献[12]中使用多节点进化神经网络进行深度学习(MENNDL)，以找到最佳的卷积掩模大小和滤波器数量。然而，MENNDL 和所有 PSO 方法还会训练每个体系结构来计算适应度函数，直到找到一个好的参数为止，这无疑增大了计算量。

各种演化方法通过训练 CNN 架构来估计 CNN 的最佳参数，以便计算适应度函数。本文的目标是获得简单的 CNN 体系结构来执行分类任务，并使用数据集进行验证。本文方法在实验中多种情况下表现出了与深度网络类似的性能。简单的体系结构能够使处理成本更低，还能降低过度拟合的产生。

## 2　基于自适应假近邻方法的卷积神经网络架构

### 2.1　卷积神经网络

CNN 是一种深度学习算法，用于处理信号、图像和视频等多维数据，即使在存在噪声、移位、缩放等情况下也可以提取相关特征数据。CNN 是一个多层网络，每一层由负责不同类型操作的单元组成，卷积被用来识别局部循环特征。

典型的 CNN 体系结构具有以下顺序：a)第一层代表输入数据并且不完成任何操作(作为验证函数工作)；b)第二层负责用激活函数计算卷积操作；c)第三层通过合并操作进行二次抽样；层 b)和 c)可以重复；d)最后一层产生输出特征。

方程(1)定义了卷积运算，其中 $I$ 是尺寸为 $q \times p$ 的二维输入数据(通常为输入图像)，$\Theta$ 是尺寸为 $m \times n$ 的卷积掩模(或核)，并且 $\phi_{i,j}$ 是由区域卷积产生的值。

$$\phi_{i,j} = (I \otimes \Theta)_{i,j} = \sum_{x=-m/2}^{m/2} \sum_{y=-n/2}^{n/2} I_{i-x,j-y} \Theta_{x,y} \tag{1}$$

在式(1)中定义的卷积步骤可以写成一个 $m \times n$ 维空间中的点积。令 $\vec{v}_{i,j} = [I_{i-x_1,j-y_1}, \cdots, I_{i,j}, \cdots, I_{i-x_m,j-y_n}]$ 是对应于 $I$ 的局部区域的矢量，令 $\vec{\Theta} = [\theta_{x_1,y_1}, \cdots, \theta_{x_m,y_n}]$ 是表示卷积掩模的向量，它是通过连接 $\Theta$ 列生成的。本文假定 $\Theta$ 中元素的索引在范围 $x = -m/2, \cdots, m/2$ 和 $y = -n/2, \cdots, n/2$ 中。使用这种表示法，式(1)可表示为点积形式：

$$\phi_{i,j} = \vec{\Theta}^T \vec{v}_{i,j} \tag{2}$$

本文假设一个具有 l 个单位(或神经元)的卷积层，其中每个单位对应一个 $m \times n$ 卷积掩模 $\Theta_k$，$k = 1, \ldots, l$。使用先前描述的向量表示并固定 $I$ 的局部区域 $\vec{v}_{i,j}$，本文可以将每个具有 $\vec{v}_{i,j}$ 的 $\vec{\Theta}_k$ 的卷积步骤写为以下矩阵向量积：

$$\begin{bmatrix} \vec{\Theta}_1^T \\ \vdots \\ \vec{\Theta}_l^T \end{bmatrix} \vec{v}_{i,j} = \begin{bmatrix} \phi_{i,j}^1 \\ \vdots \\ \phi_{i,j}^l \end{bmatrix} = \vec{\phi}_{i,j} \tag{3}$$

换句话说，卷积步骤对应于线性变换 $T(\vec{v}_{i,j}): \mathbb{R}^{m,n} \to \mathbb{R}^l$，其从 $\mathbb{R}^{m \times n}$ 到 $\mathbb{R}^l$ 获取向量。

### 2.2　矩池化

近年来，矩特征在计算机视觉、模式识别、图像分类等方面的应用越来越广泛。在文献[13]中，Rubble 提出利用矩特征来估计提取二值特征时主方向，以提高特征的鲁棒性。本文将中心矩的思想应用在 CNN 的池化中，同时，采用一阶矩阵以减少 CNN 模型的复杂性。

灰度矩阵如下所示：

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{4}$$

其中: $I(x,y)$ 代表图像坐标为 $(x,y)$ 的灰度值，并且 $p,q$ 的值均为 1。

中心矩阵如下所示：

$$c = \left[ \frac{m_{10}}{m_{00}} \frac{m_{01}}{m_{00}} \right] \tag{5}$$

由以上两个等式可以得到池化域的中心矩 $c(x,y)$，通常，$c$ 是一个浮点值，它并不指向某个特定的离散值。如图 1 所

示，$(Q_{11},Q_{12},Q_{21},Q_{22})$ 表示 $c$ 的上下边界的定点，它们的位置是固定的，可以把它们叫做 $c$ 的四邻域。本文利用插值法计算其响应值。

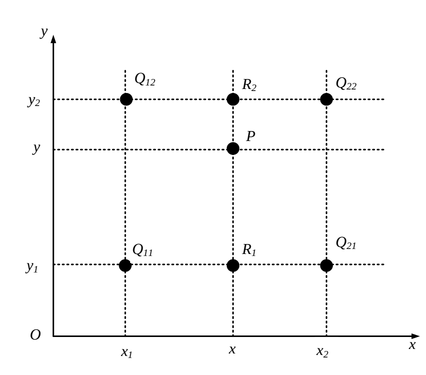



图 1　矩池化选择方法

$$\begin{aligned} & x_1=(\text{int})x, y_1=(\text{int})y \\ & x_2=x_1+1, y_2=y_1+1 \\ & Q_1=(x_1,y_1) \\ & Q_2=(x_2,y_1) \\ & Q_3=(x_2,y_2) \\ & Q_4=(x_1,y_2) \end{aligned} \tag{6}$$

常用的插值法有两种，最近邻插值法差值速度快，但该方法在使用过程中易造成棋盘格效应；双线性插值法即分别在横纵坐标轴上进行线性差值，以权值的方式衡量距离待插入点的远近，该算法有可能会使轮廓信息变得模糊，且计算量也较大。

本文根据概率 $p_x, p_y$，在横纵坐标轴上随机的选择 $x, y$，距离中心矩越近时，被选中的概率就越大，如式(7)(8)所示，再根据 $x, y$ 的值，从四邻域中选择池化区域的响应值。随机选择可以使结果避免出现过抑合的效果，且计算复杂度也较低。

$$\begin{aligned} & p_1=(x_2-x) \\ & p_2=(y_2-y) \end{aligned} \tag{7}$$

$$\begin{aligned} & x=\begin{cases} x_1 & p_x \le p_1 \\ x_2 & p_x > p_1 \end{cases} \\ & y=\begin{cases} y_1 & p_y \le p_1 \\ y_2 & p_y > p_1 \end{cases} \end{aligned} \tag{8}$$

## 2.3　无监督的训练

在 CNN 的实际应用中，一个难点是如何获取大量的有标签训练数据，而数据训练的无监督化正好可以解决此问题。我们可以利用稀疏滤波( sparse filtering)算法，即构建特征分布矩阵求解，并利用归一化的方法对样本特征实行处理，就能够得到样本分布矩阵，该矩阵的样本分布、激活时间都具有稀疏性。此外，还能实现样本分布矩阵的级联以形成无监督学习模型。

令 $F=S^TX$ 为构建的特征分布矩阵，其中 $X=[x_1,x_2,\cdots,x_n]\in \mathrm{R}^{d\times n}$ 表示训练样本矩阵。$F\in \mathrm{R}^{t\times n}$ 表示 $X$ 的特征分布矩阵，它的每一行都表示一种特征，每一列代表一个投影后的训练样本，$S$ 表示投影矩阵。稀疏滤波算法的目的即求出 $S$ 的最优解，使 $F$ 满足提及的 3 种特性。

假设 $f_{(i,\Delta)}\in R^{l\times n}$ 为 $F$ 的第 $i$ 行，$f_{(\Delta,j)}\in R^{t\times l}$ 为 $F$ 的第 $j$ 列，对 $F$ 的行和列实行二范数归一化：

$$\tilde{f}_{(i,\Delta)}=f_{(i,\Delta)}/\|f_{(i,\Delta)}\|_2 \tag{9}$$

$$\tilde{f}_{(\Delta,i)}=f_{(\Delta,i)}/\|f_{(\Delta,i)}\|_2 \tag{10}$$

对 $F$ 进行稀疏优化求解，就能够得到投影矩阵 $S$：

$$\min_s \|\hat{F}\|_1=\min_s \sum_{j=1}^{n}\|\hat{f}_{(\Delta,j)}\|_1=\sum_{j=1}^{n}\left\|\frac{\tilde{f}_{(\Delta,j)}}{\|\tilde{f}_{(\Delta,j)}\|_2}\right\|_1 \tag{11}$$

将样本分布矩阵进行级联，并将 $S$ 看作卷积核，就能得到无监督学习模型，这种结构模型在训练的过程中使用贪婪算法逐层求解，只有在当前层的优化结束后，才进行下一层次的优化求解。

## 2.4　假近邻

假近临(FNN)方法是一种将时间相关数据展开为多维空间的方法，也称为相空间，能够简化建模和预测等任务。它迭代地增加相空间的维度，在每个步骤测量邻域关系变化。FNN 首先根据欧几里德距离 $D_m^2(i,r)$，将原始数据集重建为 $m$ 维(通常为 $m=2$)和固定时间延迟 $d$ (通常为 $d=1$)的空间，其中状态 $X_i(m,d)$ 的第 $r$ 个最近邻居为 $x_i^r(m,d)$。在增加一个新的维度之后，FNN 根据等式(12)计算最近状态的距离变化。如果 $V_{i,r}>rt$，则状态被视为虚假邻居，$rt$ 是用户定义的阈值。

$$V_{i,r}=\sqrt{\frac{D_{m+1}^2(i,r)-D_m^2(i,r)}{D_m^2(i,r)}}=\frac{|x_i(m,d)-x_i^r(m,d)|}{D_m(i,r)} \tag{12}$$

当数据包含噪声或样本不够大时，此准则会计算错误的估计值。此时，可以根据式(13)避免该现象的发生，其中 $D_{m+1}^2(i)$ 为对应于邻居之间的距离，$D_{m+1}(i,r)$ 表示给定时间延迟 $\tau$ 的邻居之间的距离，$D_A^2$ 是每个点和所有其他点之间的平均距离，$at$ 是用户定义的阈值。

$$\frac{D_m(i)+D_{m+1}(i+\tau,r+\tau)}{D_A^2}>at \tag{13}$$

最后，FNN 计算整个数据集的假邻居的分数以提供截止点，从而定义嵌入维数 $m$。

## 2.5　自适应假近邻方法

CNN 在某些图像 $I$ 的局部区域中使用的向量表示 $\vec{v}_{i,j}=[I_{i-x_1,j-y_1},\cdots,I_{i,j},\cdots,I_{i-x_m,j-y_n}]$是在考虑两个嵌入维度：沿着

行($M$)的嵌入维数和在列($N$)上应用嵌入维度之后获得的，它们一起定义矢量$\vec{v}_{i,j}$。因此，为了正确估计给定 CNN 层的卷积掩模大小，本文调整了假近邻(FNN)方法。

对于$i=1,\ldots,q$和$j=1,\ldots,p$，向量$\vec{v}_{i,j}$组成了集合$\mathcal{I}=\{\vec{v}_{1,1},\ldots,\vec{v}_{q,p}\}$，对应的嵌入维度为$\mathbb{R}^{M\times N}$。本文使用 FNN 来估计卷积掩模的理想维度，如算法 1 中所述，其需要最近邻居的数目($nn$)作为输入，本文设置$nn=1$。参数$at$设置一个阈值来表征两个向量是否为假邻居。参数$\tau$定义了本文预期会发生复发的偏移量。本文期望输入数据的附近区域保持与本文实验中考虑的图像数据集相似的特征，因此设置$\tau=1$。

作为输出，算法 1 提供输入空间$\mathcal{I}$的假最近邻的平均值，在动态系统的情况下，与该组的递归估计相同。通过维持邻域，使$\mathcal{I}$中的向量与相空间中形成的吸引子的一组状态的行为相同，这意味着重建的空间需要一组更简单的函数来提供。因此，不良形式的嵌入会导致算法产生较大的输出值。当达到足够好的嵌入时，虚假的最近邻居测量应该在较低的值处并变得稳定。

算法 1 相空间重构后分析矢量集$\mathcal{I}$的假近邻值

---

数据：$\mathcal{I}$：给定数据$I$的向量集合$\vec{v}_{i,j}$；

　　$nn$：最近邻数量；

　　$at$：邻域的阈值半径；

　　$\tau$：$\mathcal{I}$中得到的向量偏置。

结果：$ffn$：假邻点的值；

设置 $counter=0$；

设置 $denom=0$；

对于所有矢量$\vec{v}_{i,j}\in\mathcal{I}$，执行：

　　找到$\mathcal{I}\{\vec{v}_{idx}\}$中最接近$nn$的矢量

　　对于每个$\vec{v}_{idx}$中的邻居$\vec{n}_{idx}$，执行

　　　　$d=Euclidean(\vec{n}_{idx},\vec{v}_{idx})$

　　　　$d=d+Euclidean(\vec{n}_{idx+\tau},\vec{v}_{idx+\tau})/d$

　　　　如果$d>at$则

　　　　　　$counter++$

　　　　结束

　　结束

　　$denom=denom+nn$

结束

$ffn=counter/denom$

---

算法 2 通过迭代改变它们的行数$M$和列$N$来评估不同的卷积掩模大小。实际上，它将掩模行的数量从 1 改变为$maxM$(最大行数)，并将掩模列从 1 改为$maxN$(最大列数)。对于每个掩模大小，输入数据$I$被嵌入多维“相位空间”中，其中计算假最近邻值的结果存储在矩阵$F$中，矩阵$F$的条目与评估的每个卷积掩模的行数$M$和列数$N$相对应。

算法 2 卷积单元中掩模大小的估计算法

---

数据：输入数据$I$；

　　$maxM$和$maxN$分别为一个假定卷积掩模中行和列的最大值；

结果：$F$，包含对于所有评估的掩模大小中假近邻值的矩阵。

初始化具有$maxM$和$maxN$的矩阵$F$；

对于$M=1$到$maxM$，执行：

　　对于$N=1$到$maxN$，执行：

　　　　$\mathcal{I}=embedd(I,M,N)$

　　　　$F[M,N]=FNNAnalysis(\mathcal{I},nn,at,\tau)$

　　结束

结束

---

本文还采用了相同的方法来研究每个 CNN 层的卷积单元的数量。将$I$的一个区域$\vec{v}_{i,j}$与所有单位掩模进行卷积，相当于将$\vec{v}_{i,j}$映射到一维空间，其中$I$是卷积单元的数量。很明显，当$I>M\times N$时，本文增加了底层空间的维度。当$I=M\times N$时，本文可以进行逆转换，简化训练阶段的方程。当$I<M\times N$时，由于维数减少，本文最终将输入空间中的不同向量映射到输出空间中的同一个向量中。因此，本文应该寻找$I\leq M\times N$的空间。

本文使用算法 3 来评估卷积单元在该线性变换中增加时的影响。该新算法接收输入数据$I$，掩模大小(行数$M$和列数$N$)，最近邻居数$nn$，阈值$at$，偏移量$\tau$，以及最后要考虑的卷积单元$l$的最大数量。作为输出，该算法提供在感兴趣层上存在的 1 到$l$个单位的 FNN 测量结果。预计在添加卷积单元时虚假邻居的数量会减少，在这种情况下，本文估计每个 CNN 层的单位数量作为向量$U$的指数，其首先达到零(或接近)用于 FNN 测量，以尝试定义最适当的维度。

算法 3 估计给定 CNN 层中卷积单元数量的算法

---

数据：输入数据$I$；

　　$M$和$N$：分别为每个卷积掩模中行和列；

　　$nn$：最近邻数量；

　　$at$：邻域的阈值半径；

　　$\tau$：$\mathcal{I}$中得到的向量偏置；

　　$l$：考虑使用的单元最大值。

结果：$U$：包含每一个添加到卷积层时假近邻值的向量。

初始化具有$l$的矩阵$U$；

初始化矩阵$\Theta$为空；

$\mathcal{I}=embedd(I,M,N)$

对于$k=1$到$l$，执行：

　　生成一个$M\times N$的随机卷积$\Theta_k$

　　将$\vec{\Theta}_k^T$作为矩阵$\Theta$的下一行

　　将矢量$\mathcal{I}$上的卷积应用于窗体中以获得新的空间$S$，且$S=(\Theta\mathcal{I}^T)^T$

　　$U[k]=FNNAnalysis(S,nn,t,\tau)$

结束

---

## 3　实验分析

### 3.1　数据集

本文使用三个数据集进行实验，所有数据集被分成训练集和测试集，训练集通过本文的 FNN 方法估计参数，其不需要标签信息，图 2 为数据集的图像样本。

a）CMU 人脸图像数据集[14]。这个图像数据集包含 1872 个灰度的人脸图像，包含四个变体：姿势(左、右、上和下)、表情(快乐、悲伤、愤怒和中性)、眼睛(不戴太阳镜)和尺寸(128×120、64×60 和 32×30)。使用构成 640 个例子的最小图像和两个不同类别进行实验，将数据集分成训练组和测试组，前者由 70%的样本组成，后者由剩下的 30%组成。

b）MNIST 数据集[15]。这是一个手写数字二进制图像的数据集，由 60000 个训练样本和 10000 个测试样本组成。

c）COIL-100[16]。哥伦比亚大学图像库 COIL-100 是一个包含 100 个类的数据集，拥有 7200 个 RGB 图像(每个类有 355 个图像，并具有姿态变化)。本文将图像调整为 32×32 像素，以便更快地进行实验，其中，70％的样本用于训练，其余 30％用于测试。



CMU人脸图像数据集

MNIST数据集



COIL-100图像集

图 2　数据集的图像样本

### 3.2　设置

自适应 FNN 的参数设置如下：最近邻居的数量 $nn=1$，邻居的半径阈值 $at=10$；对于算法 1-3，算法 1 中包含的向量的偏移量 $\tau=1$；算法 2 的最大行数和最大列数被设置为图像的大小除以 2，因此 $M$ 和 $N$ 根据分析中的数据集的最佳掩模大小来设置；算法 3 的最大卷积单元数 $l=50$。

SGD 方法用于训练所有基于 Caffe 的 CNN。本文使用卷积、ReLU 和 max-pooling 组成连续图层。Max-pooling 应用于每 3×3 像素的 ReLU 层后，步长为 2×2。训练阶段所用参数设置为：学习率为 0.001(固定-无减少)，动量为 0.9，训练批量为 100 个样本。求解器参数是凭经验选择的，可以评估卷积掩模尺寸和每层单位数量的影响。

### 3.3　参数评估

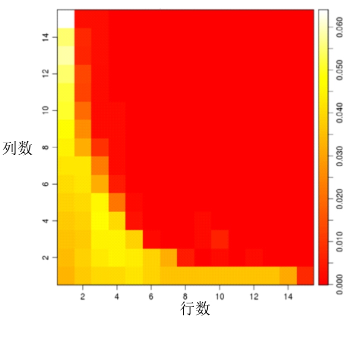
本文对训练集中的单个图像执行算法 2，以提供 FNN 测量的矩阵 $F$，其改变掩模大小的行数和列数。本文在计算 FNN 度量值时计算这些矩阵的平均值，然后使用此平均矩阵的范数来制定停止标准。如果当前和前一个标准之间的差异小于用户定义的阈值(此处设置为 0.001)，则停止执行。最后，从平均矩阵 $F$ 计算直方图以选择最佳的卷积掩模尺寸。因此，对于矩阵 $F$ 中的每一行，计数呈现第一个零或最近邻居分数的最小值的列。接下来，沿着列进行同样的处理，即对于每列，计算具有零或最小值的行的邻居分数。直方图由一个矩阵表示，该矩阵指示哪一行和哪一列最适合，其最大值为 2，最小值为零。

考虑每个数据集的最佳掩模大小。本文在给定的训练集中的每个输入图像上运行算法 3，以提供 FNN 测量的向量 $U$，同时改变单层的卷积单元数。因此，单元数由含有第一个零(或尽可能最低的值)的向量 $U$ 的索引给出，由平均 FNN 测量产生。这个零意味着线性变换的目标空间足以表示输入空间。

本文利用 MNIST 数据集来计算本文方法在估计掩模尺寸和每层单元数量时的结果。在掩模尺寸估计的情况下，将来自训练集的随机图像当作输入，直到达到停止标准(当前和前面的平均矩阵范数之间的差小于阈值时)。图 3 给出了训练图像及其对应的 FNN 矩阵，其中，(a)表示属于训练集的样本，(b)表示对应于(a)中的样本产生的假最近邻分数的矩阵，$x$ 轴表示按列表示的掩模大小，$y$ 轴表示掩模行数。使用 FNN 矩阵来计算基于元素的平均矩阵。最后，在这个最终矩阵上制作一个直方图，以返回最佳的卷积掩模大小。



(a)




(b)

图 3　利用 FNN 方法在输入图像中获得的样本结果

表 1 给出了用于估计单元数量的单元平均矢量，该结果也是从 MNIST 数据集获得的，考虑单个卷积层和掩模大小为 8×10，单位数量选择的值是这样一个向量中第一个零的索引。

表 1　MNIST 数据集上单层 CNN 的单元数量估计结果

| 序号 | FNN 比例 |
|---|---|
| 1 | 1 |
| 2 | 0.145369373 |
| 3 | 0.019133798 |
| 4 | 0.008968954 |
| 5 | 0.005095382 |
| 6 | 0.002178035 |
| 7 | 0.000692411 |
| 8 | 0.000716954 |
| 9 | 0 |
| 10 | 0 |

考虑到单卷积层和最大汇集层，对上述三个数据集都进行了此过程。接下来，利用 Caffe 深度学习框架[17]中的 SGD 进行数据训练，并且在最大共享层之后产生的输出图像被用于研究第二层卷积。再次执行该过程以添加第三层卷积。最终的网络由三个卷积层和最大池层组成。

表 2 列出了几种选定的卷积掩模尺寸及其各自单元数量的误差率(ER)，此深度学习框架是通过 10000 次迭代执行的，以选择最佳的卷积掩模大小。

### 3.4　使用最佳的掩模尺寸和卷积单位数量

在研究了最佳的掩模尺寸和卷积单元的数量后，采用最佳参数的实验对本文 CNN 架构进行 50000 次迭代训练。表 3 给出了所有数据集的错误率、相应的掩模大小和使用本文的 FNN 方法选择的卷积单元的数量。由表 3 可知，本文的 FNN 方法即使使用一小部分实例来评估掩模大小和单位数量也能提供良好的结果。

表 2　三个数据集的实验结果

| 数据集 | 1 卷积层 | | | 2 卷积层 | | | 3 卷积层 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 掩模 | 单元 | ER | 掩模 | 单元 | ER | 掩模 | 单元 | ER |
| | 8 × 1 | 12 | 0.96% | 9 × 1 | 12 | 1.16% | 4 × 2 | 18 | 3.09% |
| CMU | 5 × 2 | 13 | 0.56% | 7 × 2 | 10 | 1.62% | 6 × 4 | 5 | 2.01% |
| | 4 × 4 | 10 | 1.09% | 5 × 3 | 14 | 1.15% | 2 × 6 | 8 | 2.55% |
| | 12 × 9 | 7 | 2.27% | 8 × 4 | 8 | 2.53% | 6 × 3 | 5 | 2.95% |
| MNIST | 8 × 10 | 9 | 1.85% | 7 × 5 | 7 | 2.25% | 5 × 4 | 5 | 3.02% |
| | 11 × 11 | 8 | 1.92% | 6 × 6 | 6 | 1.88% | 2 × 5 | 6 | 2.86% |
| | 15 × 2 | 13 | 0.66% | 6 × 3 | 13 | 0.40% | 3 × 3 | 8 | 0.82% |
| COIL-100 | 13 × 3 | 15 | 0.58% | 5 × 7 | 17 | 0.51% | 2 × 4 | 11 | 0.64% |
| | 12 × 5 | 14 | 0.51% | 10 × 1 | 32 | 0.36% | 5 × 1 | 24 | 0.59% |

表 3　使用本文 FNN 方法提供的掩模大小和卷积单元数时获得的实验结果

| 数据集 | 1 卷积层 | | | 2 卷积层 | | | 3 卷积层 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 掩模 | 单元 | ER | 掩模 | 单元 | ER | 掩模 | 单元 | ER |
| CMU | 5 × 2 | 13 | 0.56% | 9 × 1 | 12 | 1.16% | 7 × 2 | 10 | 1.62% |
| MNIST | 8 × 10 | 9 | 1.85% | 6 × 6 | 6 | 1.88% | 5 × 4 | 5 | 3.02% |
| COIL-100 | 12 × 5 | 14 | 0.51% | 10 × 1 | 32 | 0.36% | 3 × 3 | 8 | 0.82% |

对于 CMU 人脸图像数据集，本文针对图像中的人进行分类，而本文的方法在单个卷积层中取得的最低误差率为 0.56%。本文的方法有助于找到最佳线性变换，以便根据共享域中向量的邻域提供相关特征。使用这个数据集的结果表明本文的策略对本文所要求的 CNN 体系结构进行配置很有用。

在文献[11]的方案中，MNIST 数据集获得的最低(最佳)错误率为 0.23%。考虑到单个卷积层，本文的方法实现了错误率为 1.85%，其中两个卷积层为 1.88%、三个卷积层为 3.02%，这是较差的，因为本文的实验使用更简单的 CNN 架构来执行，而文献[11]中使用了一个非常复杂的结构，它包含 35 个 CNN(MCDNN)，并具有以下层序列：第一个卷积层中有 20 个单元；二次抽样层中有 20 个单位；第二卷积层中有 40 个单元；下一个子采样层中有 40 个单位；多层感知器(MLP)隐藏层中有 150 个单元；MLP 的另一个隐藏层中有 10 个单元。

基于这种比较可以得出结论，本文的方法支持设计更简单的架构，在适当的参数下也可以产生良好的结果。此外，此评估强化了在尝试表示输入数据时选择足够的掩模大小和每个 CNN 层的卷积单元数量的相关性。另外，增加新的卷积层不会如预期那样影响结果。

文献[18]中使用 100 个深度卷积小波网络(DCWN)，在 COIL-100 数据集上获得的错误率等于 0.8%，而本文的方法在 CNN 中使用两个卷积层获得的错误率为 0.36%。本文结果较优是因为对此类数据集使用均衡滤波器进行了处理。

由上述分析可以得出结论：本文的 FNN 方法有助于设计更

简单的 CNN 体系结构来处理不同的分类任务，同时产生较好的分类结果。

表 4 给出了本文的方法与其他方法的错误率比较，采用了本文中使用的三个数据集。表中展示的是实验得到的最好结果，其中具有单个卷积层的体系结构由 1C 表示，两个卷积层由 2C 表示，三个卷积层由 3C 表示。

表 4　本文方法与其他研究的错误率比较

| 数据集 | 方案 | 错误率 |
|---|---|---|
| CMU | 本文方案 (1C) | 0.56% |
| | 文献[19]中方案 (遗传序列) | 7.36% |
| MNIST | 文献[11]中方案(MCDNN) | 0.28% |
| | 文献[12]中方案 (Embed CNN) | 1.25% |
| | 本文方案(3C) | 3.02% |
| COIL-100 | 本文方案 (2C) | 0.36% |
| | 文献[18]中方案 (DCWN) | 0.81% |
| | 文献[12]中方案 (Embed CNN) | 7.82% |

## 4　结束语

本文提出了一种新方法来估计卷积掩模大小和为某个目标分类任务设计简单 CNN 所需的单元数量。本文的方法基于动态系统背景下的假最近邻(FNN)技术来估计足够小的嵌入维以执行任务简化，将相关数据转化到多维空间，利用稀疏滤波算法实现训练过程的无监督化。本文不考虑时间依赖性，注重的是数据(图像)重现。本文的 FNN 方法模拟了相位空间中掩模尺寸的重现结果。本文还展示了 CNN 卷积层对输入数据应用线性变换以产生给定的共享域，这种共享域中的向量邻域是由本文的策略评估的。本文的 FNN 方法也被用来估计构成简单 CNN 的卷积单元的数量。实验证实，本文的 FNN 方法在设计更简单的 CNN 架构时支持适当的参数化，为分类任务找到好的参数，并且它需要较少的层次来设计 CNN。

## 参考文献：


[1] 周文杰，严建峰，杨璐. 基于深度学习的用户投诉预测模型研究 [J]. 计算机应用研究，2017, 34 (5): 1428-1432. (Zhou Wenjie, Yan Jianfeng, Yang Lu. Research on prediction model of complaint based on deep learning [J]. Application Research of Computers, 2017, 34 (5): 1428-1432. )

[2] Chen Yushi, Lin Zhouhan, Zhao Xing, *et al.* Deep learning-based classification of hyperspectral data [J]. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, 2017, 7 (6): 2094-2107.

[3] 田壮壮，占荣辉，胡杰民，等. 基于卷积神经网络的 SAR 图像目标识别研究 [J]. 雷达学报，2016, 5 (3): 320-325. (Tian Zhuangzhuang, Zhan Ronghui, Hu Jiemin, *et al.* SAR ATR based on convolutional neural network [J]. Journal of Radars, 2016, 5 (3): 320-325. )

[4] Gulcehre C, Firat O, Xu K, *et al.* On integrating a language model into neural machine translation [J]. Computer Speech & Language, 2017, 45 (C): 137-148.

[5] 李旭冬，叶茂，李涛. 基于卷积神经网络的目标检测研究综述 [J]. 计算机应用研究，2017, 34 (10): 2881-2886. (Li Xudong, Ye Mao, Li Tao. Review of object detection based on convolutional neural networks [J]. Application Research of Computers, 2017, 34 (10): 2881-2886. )

[6] Lauer F, Suen C, Bloch G. A trainable feature extractor for handwritten digit recognition [J]. Pattern Recognition, 2007, 40 (6): 1816-1824.

[7] Lin Weichao, Ke S W, Tsai C F. CANN: an intrusion detection system based on combining cluster centers and nearest neighbors [J]. Knowledge-Based Systems, 2015, 78 (1): 13-21.

[8] Goodfellow I, Warde-Farley D, Mirza M, *et al.* Maxout networks [J/OL]. https://arxiv. org/pdf/1302. 4389. pdf.

[9] Pierre Sermanet, Yann Lecun. Traffic sign recognition with multi-scale Convolutional Networks [C]// Proc of International Joint Conference on Neural Networks, 2011: 2809-2813.

[10] Yang M H, Dan R, Ahuja N. Learning to recognize 3D objects with SNoW [C]// Proc of European Conference on Computer Vision. 2000: 439-454.

[11] Albelwi S, Mahmood A. A framework for designing the architectures of deep convolutional neural networks [J]. Entropy, 2017, 19 (6) , 239-242.

[12] Young S R, Rose D, Karnowski T, *et al.* Optimizing deep learning hyper-parameters through an evolutionary algorithm [C]// Proc of Workshop on Machine Learning in High-Performance Computing Environments. 2015: 4-7.

[13] Rublee E, Rabaud V, Konolige K, *et al.* ORB: an efficient alternative to SIFT or SURF [C]// Proc of International Conference on Computer Vision. 2011: 2564-2571.

[14] Shotton J, Fitzgibbon A, Cook M, *et al.* Real-time human pose recognition in parts from single depth images [J]. Communications of the ACM, 2013, 56 (1): 116-124.

[15] Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000, 290 (5500): 2323-2326.

[16] LeCun Y, Bottou L, Haffner P. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86 (11): 2278-2324.

[17] Youness C, Asnaoui K, Ouanan M, *et al.* CBIR using the 2-D ESPRIT method: application to Coil_100 database [C]// Computer Systems and Applications, 2016: 1-8.

[18] Hassairi S, Ejbali R, Zaied M. Supervised image classification using deep convolutional wavelets network [C]// Proc of International Conference on Tools with Artificial Intelligence, 2016: 265-271.

[19] Zhu Chenchen, Zheng Yutong, Khoa L, *et al.* Weakly supervised facial analysis with dense hyper-column features [C]// Proc of Computer Vision and Pattern Recognition Workshops, 2016: 93-101.